

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Appellant:	EZZAT	Patent Application	
Application No.:	10/769,594	Group Art Unit:	2135
Filed:	January 30, 2004	Examiner:	Gyorfi, Thomas A.

For: PROVIDING A FLEXIBLE PROTECTION MODEL IN A COMPUTER SYSTEM  
BY DECOUPLING PROTECTION FROM COMPUTER PRIVILEGE LEVEL

APPEAL BRIEF

## Table of Contents

	<u>Page</u>
Real Party in Interest	1
Related Appeals and Interferences	2
Status of Claims	3
Status of Amendments	4
Summary of Claimed Subject Matter	5
Grounds of Rejection to Be Reviewed on Appeal	10
Argument	11
Conclusion	18
Appendix – Clean Copy of Claims on Appeal	19
Appendix – Evidence Appendix	26
Appendix – Related Proceedings Appendix	27

I. Real Party in Interest

The assignee of the present application is Hewlett-Packard Development Company,  
L.P.

## II. Related Appeals and Interferences

There are no related appeals or interferences known to the Appellant.

### III. Status of Claims

Claims 1-25 are pending. Claims 22-25 are objected to. Claims 1-25 are rejected.

This Appeal involves Claims 1-25.

#### IV. Status of Amendments

All proposed amendments have been entered. An amendment subsequent to the Final Action has not been filed.

## V. Summary of Claimed Subject Matter

Independent Claims 1, 10, 12, 15 and 17 of the present application pertain to providing flexible protection in a computer system by decoupling protection from privilege.

Independent Claim 1 recites, “A method 300 of providing flexible protection in a computer system by decoupling protection from privilege (240B, 250B).” This embodiment is described at least at paragraph 0028 on page 7; paragraphs 0030-0045 on pages 8-11; Figures 2A, 2B; and Figures 3A, 3B. “Enabling receipt of information describing two or more types of protection,” is described at least at Figure 2A; step 302, Figure 3A; paragraphs 0072-0075 on pages 16 and 17; lines 6-9 of paragraph 0063 on page 15; and paragraph 0066 on page 15. “Enabling receipt of information describing a relationship between said two or more types of protection and portions of code C1B-C9B that are executed in a same privilege level 250B of the computer system 200B, wherein said relationship is not required to be linear,” is described at least at Figure 2A; Figure 2B; step 302, Figure 3A; and paragraph 0072 on page 16. “Enabling the association of said information describing said two or more types of protection and said information describing said relationship with said portions of code C1B-C9B, wherein a first portion of code allowing a second portion of code to access the first portion of code does not depend on the second portion of code allowing the first portion of code to access the second portion of code,” is described at least at Figure 2A; steps 306-320, Figures 3A, 3B; paragraphs 0076-0082 on pages 17 and 19; and last 4 lines of paragraph 0053 on page 13.

Independent Claim 10 recites, “A method 300 of providing flexible protection in a computer system by decoupling protection from privilege (240B, 250B).” This embodiment is described at least at paragraph 0028 on page 7; paragraphs 0030-0045 on pages 8-11;

Figures 2A, 2B; and Figures 3A, 3B. “Detecting a request from a first portion of code to access a second portion of code, wherein said first and second portions of code are executed in a same privilege level 250B of said computer system 200B,” is described at least at steps 308, 310, Figure 3A; and paragraph 0078-0079 on page 18. “Determining whether said first portion of code is allowed to access said second portion of code based on information describing two or more types of protection and also based on information describing a relationship between said two or more types of protection and said portions of code, wherein said relationship is not required to be linear,” is described at least at step 312, Figure 3B; lines 6-9 of paragraph 0063 on page 15; and paragraph 0080 on page 18. “If said relationship specifies that said first portion of code may access said second portion of code allowing said first portion of code to access said second portion of code,” is described at least at step 316, Figure 3B; and paragraph 0081 on page 18. “Else not allowing said first portion of code to access said second portion of code,” is described at least at step 320, Figure 3B; and paragraph 0082 on page 19.

Independent Claim 12 recites, “A computer system 200A.” This embodiment is described at least at Figure 2A; and paragraphs 0030-0037 on pages 8 to 10. “A memory unit 402” is described at least at Figure 4; and paragraphs 0091-0093 on page 22. “A processor 401 coupled to the memory unit 402, the processor 401 for executing a method 300 for enforcing protection in a computer system by decoupling protection from privilege 240B, 250B,” is described at least at Figure 2A; paragraphs 0030-0037 on pages 8 to 10; Figure 4; and paragraphs 0091-0093 on page 22. “Enabling at a user interface 240A receipt of information describing two or more types of protection,” is described at least at paragraph 0033 on pages 8-9; step 302, Figure 3A; paragraphs 0072-0075 on pages 16 and 17; lines 6-9 of paragraph 0063 on page 15; paragraph 0066 on page 15; and Figure 2A. “Enabling at the user interface



240A receipt of information describing a relationship between said two or more types of protection and portions of code are executed in a same privilege level 250B of the computer system 200B, wherein said relationship is not required to be linear,” is described at least at paragraph 0033 on pages 8-9; Figure 2A; Figure 2B; step 302, Figure 3A; and paragraph 0072 on page 16. “Enabling at a link-editor 220A the association of said information describing said two or more types of protection and said information describing said relationship with said portions of code, wherein a first portion of code allowing a second portion of code to access the first portion of code does not depend on the second portion of code allowing the first portion of code to access the second portion of code,” is described at least at Figure 2A; paragraphs 0036-0037 on pages 9-10; steps 306-320, Figures 3A, 3B; paragraphs 0076-0082 on pages 17 and 19; and last 4 lines of paragraph 0053 on page 13.

Independent Claim 15 recites, “A computer system 200B.” This embodiment is described at least at Figure 2B; and paragraphs 0038-0045 on pages 10-11. “A memory unit 402” is described at least at Figure 4; and paragraphs 0091-0093 on page 22. “A processor 401 coupled to the memory unit 402, the processor 401 for executing a method 300 for providing flexible protection in a computer system 200B by decoupling protection from privilege 240B, 250B” is described at least at Figures 2A, 2B; paragraphs 0038-0045 on pages 10-11; Figure 4; paragraphs 0091-0093 on page 22; paragraph 0028 on page 7; paragraphs 0030-0045 on pages 8-11; and Figures 3A, 3B. “Detecting at a memory manager 260B a request from a first portion of code to access a second portion of code, wherein said first and second portions of code are executed in a same privilege level 250B of said computer system 200B,” is described at least at Figure 2B; paragraphs 0041-0042 on pages 10-11; steps 308, 310, Figure 3A; and paragraphs 0078-0079 on page 18. “Determining at said memory manager 260B whether said first portion of code is allowed to access said

second portion of code based on information describing two or more types of protection and also based on information describing a relationship between said two or more types of protection and said portions of code, wherein said relationship is not required to be linear,” is described at least at paragraph 0042 on pages 10-11; step 312, Figure 3B; lines 6-9 of paragraph 0063 on page 15; and paragraph 0080 on page 18. “If said relationship specifies that said first portion of code may access said second portion of code, then allowing at said memory manager said first portion of code to access said second portion of code,” is described at least at step 316, Figure 3B; and paragraph 0081 on page 18. “Else not allowing at said memory manager said first portion of code to access said second portion of code,” is described at least at step 320, Figure 3B; and paragraph 0082 on page 19.

Independent Claim 17 recites, “A computer-usable medium having computer-readable program code embodied therein for causing a computer system to perform a method 300 of providing flexible protection in a computer system 200B by decoupling protection from privilege (240B, 250B).” This embodiment is described at least at paragraph 0028 on page 7; paragraphs 0030-0045 on pages 8-11; Figures 2A, 2B; and Figures 3A, 3B.

“Enabling receipt of information describing two or more types of protection,” is described at least at Figure 2A; step 302, Figure 3A; paragraphs 0072-0075 on pages 16 and 17; lines 6-9 of paragraph 0063 on page 15; and paragraph 0066 on page 15. “Enabling receipt of information describing a relationship between said two or more types of protection and portions of code C1B-C9B that are executed in a same privilege level 250B of the computer system 200B, wherein said relationship is not required to be linear,” is described at least at Figure 2A; Figure 2B; step 302, Figure 3A; and paragraph 0072 on page 16. “Enabling the association of said information describing said two or more types of protection and said

information describing said relationship with said portions of code C1B-C9B, wherein a first portion of code allowing a second portion of code to access the first portion of code does not depend on the second portion of code allowing the first portion of code to access the second portion of code,” is described at least at Figure 2A; steps 306-320, Figures 3A, 3B; paragraphs 0076-0082 on pages 17 and 19; and last 4 lines of paragraph 0053 on page 13.

## VI. Grounds of Rejection to Be Reviewed on Appeal

1. Claims 22-25 are objected to for not being in consecutive order.
2. Claims 1-25 are rejected under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 6,125,447 by Gong, hereinafter referred to as “Gong.”

## VII. Argument

### 1. Objection to Claims

Appellant respectfully points out that a preliminary amendment was filed February 23, 2004 renumbering the claims to be in consecutive order.

### 2. Whether Claims 1-25 are Anticipated Under 35 U.S.C. §102(b) by Gong.

Appellant has reviewed the cited art and respectfully submits that the embodiments as recited in Claims 1-25 are not anticipated by Gong in view of the following rationale.

MPEP §2131 provides:

“A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.” *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). ... “The identical invention must be shown in as complete detail as is contained in the ... claim.” *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim.

Independent Claim 1 recites,

A method of providing flexible protection in a computer system by decoupling protection from privilege, the method comprising:

- enabling receipt of information describing two or more types of protection;
- enabling receipt of information describing a relationship between said two or more types of protection and portions of code that are executed in a same privilege level of the computer system, wherein said relationship is not required to be linear;
- and

- enabling the association of said information describing said two or more types of protection and said information describing said relationship with said portions of code, wherein a first portion of code allowing a second portion of code to access the first portion of code does not depend on the second portion of code allowing the first portion of code to access the second portion of code.

#### A. Cited Art does not show Each and Every Element as set Forth in the Claim

Referring to Col. 10 and Figure 2, Appellant understands Gong to teach using a code identifier 232 associated with a class 260 to determine whether an object 262 that is an

instance of that class 260 will be granted permission to access a file. For example, Col. 10 lines 47-50 states, “There is only one permission associated with the code identifier ‘file//somesource’-‘somekey’, which is a permission to write to any file in directory ‘Itmp/\*’”. From Col. 11 line 40 to Col. 13 line 22, Gong describes the execution of a thread. At Col. 11 lines 46-53, Gong states,

For example, assume that a thread executes a.x (where “a” is an object and “x” is a method associated with object “a”). Assume that a.x invokes b.y which invokes c.z. While c.z is executing, the call stack will contain data identifying a.x, b.y, and c.z. At this point, call stack 610 represents the calling hierarchy of the methods invoked by the thread but have not yet been completed by the thread.

Gong also states at Col. 11 lines 55-58,

Note that objects corresponding to the method invocations in the call stack are each associated with a protection domain. Object a is associated with protection domain I and object b and object c are associated with protection domain J.

Gong states at Col. 12 lines 7-8, “Object a is requesting to write to file ‘/tmp/temporary’.”

Gong states at Col. 12 lines 30-35,

A requested action is authorized if every protection domain associated with the objects represented by the call stack when the request for the requested action was made contains a permission authorizing the permission required to perform the requested action.

According to this example, Appellant understands Gong to deny the request to access ‘/tmp/temporary’ (Col. 13 lines 19-21) because referring to Fig. 6, although a.x maps to protection domain I, which permits write access to “/tmp/”, b.y and c.z map to protection domain J, which does not permit write access to “/tmp/.” Therefore, Appellant understands Gong to teach that portions of code (a.x, b.y, c.z) attempting to access data (/tmp/) are dependent on each other for permission to access the data.

Therefore, Appellant does not understand Gong to teach, describe or suggest “wherein a first portion of code allowing a second portion of code to access the first portion

of code does not depend on the second portion of code allowing the first portion of code to access the second portion of code,” as recited by independent Claims 1, 12 and 17.

Further, Appellant does not understand Gong to teach, describe or suggest “information describing a relationship between... two or more types of protection,” (emphasis added) as recited by independent Claims 1, 10, 12, 15, and 17. For example, Appellant does not understand Gong to teach information describing a relationship between protection domain I (permission to write to or to read from /tmp/\*\*) and protection domain J (permission to write to or to read from /share/\*\*).

Lastly, Appellant does not understand Gong to teach “portions of code ... executed in a same privilege level of said computer system,” as recited by independent Claims 1, 10, 12, 15 and 17. For example, the Office Action asserts that Gong teaches “enabling receipt of information describing a relationship between said two or more types of protection and portions of code that are executed in a same privilege level of the computer system, wherein said relationship is not required to be linear,” as recited by Claim 1 at Col. 9 lines 40-53. However, Appellant does not understand Col. 9 lines 40-53 to teach, describe or suggest “privilege levels” and therefore Appellant does not understand Gong to teach, describe or suggest, “portions of code ... executed in a same privilege level of said computer system,” as recited by independent Claims 1, 10, 12, 15 and 17.

## RESPONSE TO ARGUMENTS

The Office Action states in the Response to Arguments section on page 2,

Applicant argues that Gong fails to recite every limitation of the claim by arguing that “Gong pertains to portions of code (a.x, b.y, c.z) attempting to access data (/tmp/) where the portions of code depend on each other for accessing data.” ... Interestingly,

Applicant quoted col. 11, lines 46-53, which discloses that a.x invokes b.y, which in turn invokes c.z; however, there is no evidence in the prior art to support that a reverse relationship necessarily exists. Indeed, this passage discloses that when a.x invokes b.y, a.x is saved on the stack and remains inactive until b.y finishes its task; similarly, as b.y requires c.z, so b.y is also saved on the stack and waits for c.z to complete its task.

Appellant respectfully submits that the Office Action's statement has taken Appellant's statement out of context. Appellant stated in the Response to the Office Action dated April 19, 2007,

Gong states at Col. 12 lines 7-8, "Object a is requesting to write to file 'tmp/temporary'." Gong states at Col. 12 lines 30-35,

A Requested action is authorized if every protection domain associated with the objects represented by the call stack when the request for the requested action was made contains a permission authorizing the permission required to perform the requested action.

According to this example, the request to access '/tmp/temporary' is denied (Col. 13 lines 19-21) because referring to Fig. 6, although a.x maps to protection domain I, which permits write access to "/tmp/", b.y and c.z map to protection domain J, which do not permit write access to "/tmp/." Therefore, Gong pertains to portions of code (a.x, b.y, c.z) attempting to access data (/tmp/) where the portions of code depend on each other for accessing the data.

Appellant is pointing out that Gong's portions of code which in Gong's implementation are on a stack are depending on each other for permission to access data.

The Office Action goes on to state in the Response to Arguments,

Additionally, it is noted that access to any portion of code is permitted when any permission in any protection domain allows for it (col. 12 lines 50-53). In the specific example quoted by Gong, a.x would be able to access "/tmp/temporary" because a.x is a member of protection domain I, which allows for anyone to write to the /tmp directory (col. 13, lines 3-9); thus a.x may access it without regard to b.y or c.z. The reverse is not true, however, as neither b.y nor c.z have explicit permissions on the /tmp directory as they are members of protection domain J, which only gives permissions to the /share directory (col. 13, lines 10-22). Because the relationship regarding permissions is asymmetric as disclosed above, it therefore still reads on the amended claims.



Appellant respectfully disagrees with the Office Action's interpretation of Gong's teachings. Referring to Figure 6 and Col. 11 line 33 to Col. 13 line 22 of Gong, Appellant understands Gong to teach the following: Referring to Col. 11 lines 45-54, a thread executes a.x which invokes b.y which invokes c.z where a, b, c are objects and x, y, z are methods of the respective objects a, b, c. While c.z is executing, the call stack includes data identifying the calling hierarchy (a.x, b.y, c.z in this example) of the methods (x, y, z) invoked by the thread that have not yet been completed by the thread.

Gong states at Col. 11 line 67 to Col. 12 line 2,

Assume that a thread invokes a.x, b.y, and c.z in the manner described so that the call stack 610 is as it appears in FIG. 6. Referring to FIG. 5, assume in step 550 a request to perform a particular action is then received from the thread while thread is executing c.z. Typically, a request is in the form of an attempt to invoke a particular method that performs a particular operation. In this example, the particular request is made by object a. In other words, a method associated with object a invoked a method that may perform the particular action. Object a is requesting to write to file "/tmp/temporary". The permission required to perform this action is a "write" permission for file "/tmp/temporary". The permission required to perform a requested action is herein referred to as a required permission (emphasis added).

Referring to Col. 11 line 67 to Col. 12 line 2 quoted above, a request to perform a particular action is then received from the thread while the thread is executing c.z. Referring to Col. 12 lines 4-11 quoted above object a requests a particular action to write to file /tmp/temporary. Referring to Col. 12 lines 8-11 quoted above, the "required permission" for object a to perform the requested action is to write to file "/tmp/temporary" for the example described from Col. 11 line 33 to Col. 13 line 22.

As already discussed herein, Gong states at Col. 12 lines 30-35 that "A requested action is authorized if every protection domain associated with the objects represented by the call stack when the request for the requested action was made contains a permission

authorizing the permission required to perform the requested action” (emphasis added). Returning to the example, at this point, a.x, b.y and c.z are on the call stack. Referring to Col. 13 lines 4-9 and Figure 6, the protection domains associated with the call stack are protection domain I (“write to any file in directory /tmp”) for object a and protection domain J (“write to any file in directory /share”) for objects b and c. Therefore, the permission associated with objects b and c does not permit object a’s “required permission” to write to file /tmp/temporary. For at least this reason, Appellant understands Gong to teach that Gong’s portion of code a depends on Gong’s other portions of code b and c to access /tmp/temporary in contrast to “wherein a first portion of code allowing a second portion of code to access the first portion of code does not depend on the second portion of code allowing the first portion of code to access the second portion of code,” as recited by independent Claims 1, 12 and 17.

## SUMMARY

Appellant does not understand Gong to teach, describe or suggest “wherein a first portion of code allowing a second portion of code to access the first portion of code does not depend on the second portion of code allowing the first portion of code to access the second portion of code,” as recited by independent Claims 1, 12 and 17. Appellant does not understand Gong to teach, describe or suggest “information describing a relationship between... two or more types of protection,” (emphasis added) as recited by independent Claims 1, 10, 12, 15, and 17. Appellant does not understand Gong to teach, describe or suggest “portions of code ... executed in a same privilege level of said computer system,” as recited by independent Claims 1, 10, 12, 15 and 17.

For at least the reasons provided herein, Appellant submits that independent Claims 1, 10, 12, 15 and 17 are not anticipated by Gong as the Rejection fails to establish a *prima facie* case for anticipation of Claims 1, 10, 12, 15 and 17. As such, Appellant submits that independent Claims are in condition for allowance. Claims 2-9 depend on Claim 1. Claim 11 depends on Claim 10. Claims 13-14 depend on Claim 12. Claim 16 depends on Claim 15. Claims 18-23 depend on Claim 17. Hence, it is respectfully submitted that the dependent Claims are patentable over Gong for the reasons discussed above, and are in condition for allowance by virtue of their dependence upon a respective allowable base claim.

Conclusion

Appellant believes that pending Claims 1-25 are patentable over Gong. As such, Appellant submits that Claims 1-25 are patentable over the cited references.

Appellant respectfully request that the rejection of Claims 1-25 be reversed. The Appellant wishes to encourage the Examiner or a member of the Board of Patent Appeals to telephone the Appellant's undersigned representative if it is felt that a telephone conference could expedite prosecution.

Respectfully submitted,  
Wagner Blecher LLP

Dated: 4/18/2008

/John P. Wagner, Jr./

John P. Wagner, Jr.  
Registration No.: 35,398

Wagner Blecher LLP  
Westridge Business Park  
123 Westridge Drive  
Watsonville, CA 95076

Phone: (408) 377-0500  
Facsimile: (831) 722-2350

VIII. Appendix - Clean Copy of Claims on Appeal

1. A method of providing flexible protection in a computer system by decoupling protection from privilege, the method comprising:
  - enabling receipt of information describing two or more types of protection;
  - enabling receipt of information describing a relationship between said two or more types of protection and portions of code that are executed in a same privilege level of the computer system, wherein said relationship is not required to be linear; and
  - enabling the association of said information describing said two or more types of protection and said information describing said relationship with said portions of code, wherein a first portion of code allowing a second portion of code to access the first portion of code does not depend on the second portion of code allowing the first portion of code to access the second portion of code.
2. The method of Claim 1, wherein said relationship is user definable.
3. The method of Claim 1, wherein said portions of code are domains and each of said types of protection is defined at least in part by one or more domain attributes.
4. The method of Claim 3, wherein said one or more domain attributes includes a domain identifier that specifies to a unique value for a particular domain.

5. The method of Claim 3, wherein said one or more domain attributes includes a Private Key that specifies a unique value for protecting each user that concurrently uses a particular domain.
6. The method of Claim 3, wherein said one or more domain attributes includes a SharedCode Key that specifies a value that a particular domain must use to access code associated with another domain.
7. The method of Claim 3, wherein said one or more domain attributes includes a SharedData Key that specifies a value that a particular domain must use to access data associated with another domain.
8. The method of Claim 3, wherein said one or more domain attributes includes an AllowOthers that specifies a value that a particular domain must use to access code associated with another domain in conjunction with said particular domain performing cross-domain switching to said other domain.
9. The method of Claim 3, wherein said one or more domain attributes includes an AccessOthers Key that specifies a value that is used to request access of code associated with a particular domain on behalf of another domain.
10. A method of providing flexible protection in a computer system by decoupling protection from privilege, the method comprising:

detecting a request from a first portion of code to access a second portion of code, wherein said first and second portions of code are executed in a same privilege level of said computer system;

determining whether said first portion of code is allowed to access said second portion of code based on information describing two or more types of protection and also based on information describing a relationship between said two or more types of protection and said portions of code, wherein said relationship is not required to be linear; and

if said relationship specifies that said first portion of code may access said second portion of code, then

allowing said first portion of code to access said second portion of code;

else

not allowing said first portion of code to access said second portion of code.

11. The method of Claim 10, wherein said information describing said two or more types of protection and said information describing said relationships are associated with said portions of code and wherein the method further comprises retrieving said information describing said two or more types of protection and said information describing said relationships .

12. A computer system comprising:

a memory unit; and

a processor coupled to the memory unit, the processor for executing a method for enforcing protection in a computer system by decoupling protection from privilege, the method comprising:

200315891-1  
Serial No.: 10/769,594

Group Art Unit: 2135

enabling at a user interface receipt of information describing two or more types of protection;

enabling at the user interface receipt of information describing a relationship between said two or more types of protection and portions of code are executed in a same privilege level of the computer system, wherein said relationship is not required to be linear; and

enabling at a link-editor the association of said information describing said two or more types of protection and said information describing said relationship with said portions of code, wherein a first portion of code allowing a second portion of code to access the first portion of code does not depend on the second portion of code allowing the first portion of code to access the second portion of code.

13. The computer system of Claim 12, wherein said relationship is user definable.

14. The computer system of Claim 12, wherein said portions of code are domains and each of said types of protection is defined at least in part by one or more domain attributes.

15. A computer system comprising:

a memory unit; and

a processor coupled to the memory unit, the processor for executing a method for providing flexible protection in a computer system by decoupling protection from privilege, the method comprising:

detecting at a memory manager a request from a first portion of code to access a second portion of code, wherein said first and second portions of code are executed in a same privilege level of said computer system;



determining at said memory manager whether said first portion of code is allowed to access said second portion of code based on information describing two or more types of protection and also based on information describing a relationship between said two or more types of protection and said portions of code, wherein said relationship is not required to be linear; and

if said relationship specifies that said first portion of code may access said second portion of code, then

allowing at said memory manager said first portion of code to access  
said second portion of code;

else

not allowing at said memory manager said first portion of code to  
access said second portion of code.

16. The computer system of Claim 15, wherein said information describing said two or more types of protection and said information describing said relationships are associated with said portions of code and wherein the method further comprises retrieving at a loader said information describing said two or more types of protection and said information describing said relationships.

17. A computer-usable medium having computer-readable program code embodied therein for causing a computer system to perform a method of providing flexible protection in a computer system by decoupling protection from privilege, the method comprising:

enabling receipt of information describing two or more types of protection;

enabling receipt of information describing a relationship between said two or more types of protection and portions of code that are executed in a same privilege level of the computer system, wherein said relationship is not required to be linear; and

enabling the association of said information describing said two or more types of protection and said information describing said relationship with said portions of code, wherein a first portion of code allowing a second portion of code to access the first portion of code does not depend on the second portion of code allowing the first portion of code to access the second portion of code.

18. The computer-usable medium of Claim 17, wherein said relationship is user definable.

19. The computer-usable medium of Claim 17, wherein said portions of code are domains and each of said types of protection is defined at least in part by one or more domain attributes.

20. The computer-usable medium of Claim 19, wherein said one or more domain attributes includes a domain identifier that specifies to a unique value for a particular domain.

21. The computer-usable medium of Claim 19, wherein said one or more domain attributes includes a Private Key that specifies a unique value for protecting each user that concurrently uses a particular domain.

22. The computer-usable medium of Claim 19, wherein said one or more domain attributes includes a SharedCode Key that specifies a value that a particular domain must use to access code associated with another domain.

23. The computer-usable medium of Claim 19, wherein said one or more domain attributes includes a SharedData Key that specifies a value that a particular domain must use to access data associated with another domain.

24. The computer-usable medium of Claim 19, wherein said one or more domain attributes includes an AllowOthers that specifies a value that a particular domain must use to access code associated with another domain in conjunction with said particular domain performing cross-domain switching to said other domain.

25. The computer-usable medium of Claim 19, wherein said one or more domain attributes includes an AccessOthers Key that specifies a value that is used to request access of code associated with a particular domain on behalf of another domain.

IX. Evidence Appendix

No evidence is herein appended.

X. Related Proceedings Appendix

No related proceedings.